



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/786,786	02/24/2004	Moshe E. Matsa	POU920030086US1	3750
23334	7590	06/25/2009	EXAMINER	
FLEIT GIBBONS GUTMAN BONGINI & BIANCO P.L. ONE BOCA COMMERCE CENTER 551 NORTHWEST 77TH STREET, SUITE 111 BOCA RATON, FL 33487			LOVEL, KIMBERLY M	
		ART UNIT		PAPER NUMBER
		2167		
			NOTIFICATION DATE	DELIVERY MODE
			06/25/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

ptoboca@fggbb.com

Office Action Summary	Application No.	Applicant(s)	
	10/786,786	MATSA ET AL.	
	Examiner	Art Unit	
	KIMBERLY LOVEL	2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 30 March 2009.
 2a) This action is **FINAL**. 2b) This action is non-final.
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1,2,5-8 and 21-33 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) Claim(s) _____ is/are allowed.
 6) Claim(s) 1, 2, 5-8 and 21-33 is/are rejected.
 7) Claim(s) _____ is/are objected to.
 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ . |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____. | 6) <input type="checkbox"/> Other: _____ . |

DETAILED ACTION

Response to Amendment

1. This communication is in response to the Amendment filed 30 March 2009.
2. Claims 1, 2, 5-8 and 21-33 are currently pending and claims 3, 5 and 9-20 are canceled. In the Amendment filed 30 March 2009, claim 1 has been amended and claims 25-33 are new. This action is made Final.
3. The prior art rejections in the previous rejection have been withdrawn as necessitated by amendment.

Claim Rejections - 35 USC § 101

4. The rejections of **Claims 1, 2, 4-8 and 21-24** under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter have been withdrawn.

Claim Rejections - 35 USC § 103

5. **Claims 1, 2, 5-8 and 21-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over US PGPub 2001/0034771 to Hutsch et al (hereafter Hutsch) in view of US Patent No 6,952,703 to Kathail et al (hereafter Kathail).**

Referring to claim 1, Hutsch discloses a method for managing configuration data, the method comprising the steps of:

storing, into a computer system, a plurality of configuration values in a hierarchical tree [DOM tree] having a plurality of nodes (see [0327], lines 3-5), a defined structure [strongly typed schema] (see [0418]), and defined data types for the stored

configuration values (see [0329]), wherein the plurality of nodes includes at least one inner node [i.e., color] and at least one child node [i.e., red] that is associated with the inner node (see Fig 16A), and wherein some of the nodes are only associated with a set of configuration values while other of the nodes are associated with a combination of a set of configuration values [value] and an identifier [key] associated with at least one application component (see [0158] and [0159]);

registering, by a computer system, at least one application component with at least one of the nodes of the tree, based on at least one query [transaction] received from the at least one application component (see [0159]); and

directly notifying, by the computer system, the at least one application component [listener] when a configuration value stored in the at least one node is modified [alterations], based on an addition or change in at least one configuration value that matches the at least one query [transaction] (see [0159]).

Hutsch fails to explicitly disclose the further limitations of wherein at least one configuration value is stored in each node of the plurality of nodes and wherein the at least one node comprises at least one configuration value that dictates how the application component at least one of behaves with and interacts with other application components, wherein an application component is an entity that is defined, in part, by at least one configuration value. Kathail discloses storing configuration values in a tree, including the further limitations of wherein at least one configuration value is stored in each node of the plurality of nodes, some of the nodes are only associated with a set of configuration values while other of the nodes are associated with a combination of a set

of configuration values and an identifier (see column 6, lines 64-67) associated with at least one application component (see column 7, lines 3-27); registering at least one application component directly with at least one of the nodes of the tree (see column 8, lines 22-29), wherein the at least one node comprises at least one configuration value that dictates how the application component at least one of behaves with and interacts with other application components (see column 9, lines 41-49); and directly notifying the at least one application component when a configuration value is modified (see column 8, lines 22-36).

It would have been obvious to one of ordinary skill in the art at the time of the invention to utilize the tree and registration concepts of Kathail with the tree of Hutsch. One would have been motivated to do so in order to improve the efficiency of managing applications since it can be cumbersome to do so without a method of understanding their configuration and dependencies.

Referring to claim 2, the combination of Hutsch and Kathail (hereafter Hutsch/Kathail) discloses the method of claim 1, wherein the at least one query depends on at least one of a location of a configuration value in the tree [value] and a data type of a configuration value (Hutsch: see [0159]).

Referring to claim 5, Hutsch/Kathail discloses the method of claim 1, wherein a node further includes a reference to at least one node (Hutsch: see Fig 16A and Kathail: see column 7, lines 28-34).

Referring to claim 6, Hutsch/Kathail discloses the method of claim 1, wherein the notifying step comprises: modifying at least one configuration value that is

associated with the at least one node with which the at least one application component is registered; storing in the hierarchical tree the configuration value that was modified, the configuration value being stored at the at least one node with which the at least one application component is registered; and notifying the at least one application component that the configuration value was modified (Hutsch: see [0159] and Kathail: see column 8, lines 22-36).

Referring to claim 7, Hutsch/Kathail discloses the method of claim 6, further comprising the step of supplying the configuration value that was modified to the at least one application component (Hutsch: see [0159] and Kathail: see column 7, lines 46-53).

Referring to claim 8, Hutsch/Kathail discloses the method of claim 1, further comprising the step of supplying at least one of the configuration values stored in the hierarchical tree to the at least one application component (Hutsch: see [0159] and Kathail: see column 7, lines 46-53).

Referring to claim 21, Hutsch/Kathail discloses the method of claim 1, wherein the plurality of configuration values in the hierarchical tree includes all of the configuration data values that are required by the at least one application component (Hutsch: see [0158] and [0159] and Kathail: see column 6, line 65 – column 7, line 2).

Referring to claim 22, Hutsch/Kathail discloses the method of claim 1, wherein the step of registering at least one application component comprises registering the at least one application component with the at least one inner node (Hutsch: see [0159] and Fig 16A and Kathail: see column 9, lines 19-49 and column 11, lines 35-53).

Referring to claim 23, Hutsch/Kathail discloses the method of claim 22, wherein the step of directly notifying the at least one application component comprises directly notifying the at least one application component when at least one configuration value associated with at least one of the inner node and the child node that is associated with the inner node is modified, based on an addition or change in the at least one configuration value (Hutsch: see [0159] and Fig 16A and Kathail: see column 8, lines 30-36).

Referring to claim 24, Hutsch/Kathail discloses the method of claim 1, wherein at least one configuration value in the plurality of configuration values that is associated with a first application component overlaps with another configuration value in the plurality of configuration values that is associated with a second application component, and the at least one configuration value and the other configuration value are nested under a common sub-tree in the tree (Hutsch: see [0159] and Kathail: see column 7, lines 3-27).

Referring to claim 25, Hutsch/Kathail discloses the method of claim 1, further comprising the step of: storing at least one other configuration value in the at least one node after the at least one application component has been directly registered with at least one of the nodes, wherein directly notifying the at least one application component further comprises directly notifying the at least one application component when both the at least one configuration value and the at least one other value are modified (Kathail: see column 7, lines 20-23 and column and column 8, lines 29-36)

Referring to claim 26, Hutsch discloses a computer program product for managing configuration data, the computer program product comprising:

a storage medium readable by a processing circuit and storing instructions for execution by the processing circuit for performing a method comprising the steps of: storing, into a computer system, a plurality of configuration values in a hierarchical tree [DOM tree] having a plurality of nodes (see [0327], lines 3-5), a defined structure [strongly typed schema] (see [0418]), and defined data types for the stored configuration values (see [0329]), wherein the plurality of nodes includes at least one inner node [i.e., color] and at least one child node [i.e., red] that is associated with the inner node (see Fig 16A), and wherein some of the nodes are only associated with a set of configuration values while other of the nodes are associated with a combination of a set of configuration values [value] and an identifier [key] associated with at least one application component (see [0158] and [0159]);

registering, by a computer system, at least one application component with at least one of the nodes of the tree, based on at least one query [transaction] received from the at least one application component (see [0159]); and

directly notifying, by the computer system, the at least one application component [listener] when a configuration value stored in the at least one node is modified [alterations], based on an addition or change in at least one configuration value that matches the at least one query [transaction] (see [0159]).

Hutsch fails to explicitly disclose the further limitations of wherein at least one configuration value is stored in each node of the plurality of nodes and wherein the at

least one node comprises at least one configuration value that dictates how the application component at least one of behaves with and interacts with other application components, wherein an application component is an entity that is defined, in part, by at least one configuration value. Kathail discloses storing configuration values in a tree, including the further limitations of wherein at least one configuration value is stored in each node of the plurality of nodes, some of the nodes are only associated with a set of configuration values while other of the nodes are associated with a combination of a set of configuration values and an identifier (see column 6, lines 64-67) associated with at least one application component (see column 7, lines 3-27); registering at least one application component directly with at least one of the nodes of the tree (see column 8, lines 22-29), wherein the at least one node comprises at least one configuration value that dictates how the application component at least one of behaves with and interacts with other application components (see column 9, lines 41-49); and directly notifying the at least one application component when a configuration value is modified (see column 8, lines 22-36).

It would have been obvious to one of ordinary skill in the art at the time of the invention to utilize the tree and registration concepts of Kathail with the tree of Hutsch. One would have been motivated to do so in order to improve the efficiency of managing applications since it can be cumbersome to do so without a method of understanding their configuration and dependencies.

Referring to claim 27, Hutsch/Kathail discloses the computer program product of claim 26, wherein the at least one query depends on at least one of a location of a

configuration value in the tree [value] and a data type of a configuration value (Hutsch: see [0159]).

Referring to claim 28, Hutsch/Kathail discloses the computer program product of claim 26, wherein a node further includes a reference to at least one node (Hutsch: see Fig 16A and Kathail: see column 7, lines 28-34).

Referring to claim 29, Hutsch/Kathail discloses the computer program product of claim 26, wherein the notifying step comprises: modifying at least one configuration value that is associated with the at least one node with which the at least one application component is registered; storing in the hierarchical tree the configuration value that was modified, the configuration value being stored at the at least one node with which the at least one application component is registered; and notifying the at least one application component that the configuration value was modified (Hutsch: see [0159] and Kathail: see column 8, lines 22-36).

Referring to claim 30, Hutsch/Kathail discloses the computer program product of claim 29, further comprising the step of supplying the configuration value that was modified to the at least one application component (Hutsch: see [0159] and Kathail: see column 7, lines 46-53).

Referring to claim 31, Hutsch/Kathail discloses the computer program product of claim 26, further comprising the step of supplying at least one of the configuration values stored in the hierarchical tree to the at least one application component (Hutsch: see [0159] and Kathail: see column 7, lines 46-53).

Referring to claim 32, Hutsch discloses a computer system for managing configuration data, the computer system comprising the steps of:

an organization module for organizing a plurality of configuration values in a hierarchical tree [DOM tree] having a plurality of nodes (see [0327], lines 3-5), a defined structure [strongly typed schema] (see [0418]), and defined data types for the stored configuration values (see [0329]), wherein the plurality of nodes includes at least one inner node [i.e., color] and at least one child node [i.e., red] that is associated with the inner node (see Fig 16A), and wherein some of the nodes are only associated with a set of configuration values while other of the nodes are associated with a combination of a set of configuration values [value] and an identifier [key] associated with at least one application component (see [0158] and [0159]);

a storage medium for storing a plurality of configuration values in the hierarchical tree [DOM tree] (see [0327], lines 3-5);

a registration module for registering, by a computer system, at least one application component with at least one of the nodes of the tree, based on at least one query [transaction] received from the at least one application component (see [0159]); and

a notification module for directly notifying the at least one application component [listener] when a configuration value stored in the at least one node is modified [alterations], based on an addition or change in at least one configuration value that matches the at least one query [transaction] (see [0159]).

Hutsch fails to explicitly disclose the further limitations of wherein at least one configuration value is stored in each node of the plurality of nodes and wherein the at least one node comprises at least one configuration value that dictates how the application component at least one of behaves with and interacts with other application components, wherein an application component is an entity that is defined, in part, by at least one configuration value. Kathail discloses storing configuration values in a tree, including the further limitations of wherein at least one configuration value is stored in each node of the plurality of nodes, some of the nodes are only associated with a set of configuration values while other of the nodes are associated with a combination of a set of configuration values and an identifier (see column 6, lines 64-67) associated with at least one application component (see column 7, lines 3-27); registering at least one application component directly with at least one of the nodes of the tree (see column 8, lines 22-29), wherein the at least one node comprises at least one configuration value that dictates how the application component at least one of behaves with and interacts with other application components (see column 9, lines 41-49); and directly notifying the at least one application component when a configuration value is modified (see column 8, lines 22-36).

It would have been obvious to one of ordinary skill in the art at the time of the invention to utilize the tree and registration concepts of Kathail with the tree of Hutsch. One would have been motivated to do so in order to improve the efficiency of managing applications since it can be cumbersome to do so without a method of understanding their configuration and dependencies.

Response to Arguments

6. Applicant's argument with respect to claim 1 on pages 11-12 of the Remarks, wherein the applicant argues the newly added limitation has been considered but are moot in view of the new ground(s) of rejection. The addition of the Kathail reference has also made the other arguments moot since Kathail further discloses the further limitations. Furthermore, in regards to the argument referring to the storing of configuration values in some nodes and configuration values and an identifier of an application in other nodes, Kathail also further discloses this concept. Configuration values are stored in a tree. Only a configuration value is stored in a node unless an application has registered with the node, in which case there is a value and an associated identifier.

Conclusion

7. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the

shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Contact Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to KIMBERLY LOVEL whose telephone number is (571)272-2750. The examiner can normally be reached on 8:00 - 4:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cottingham can be reached on (571) 272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/John R. Cottingham/
Supervisory Patent Examiner, Art Unit 2167

/Kimberly Lovel/
Examiner
Art Unit 2167

22 June 2009
/KL/

Application/Control Number: 10/786,786
Art Unit: 2167

Page 15